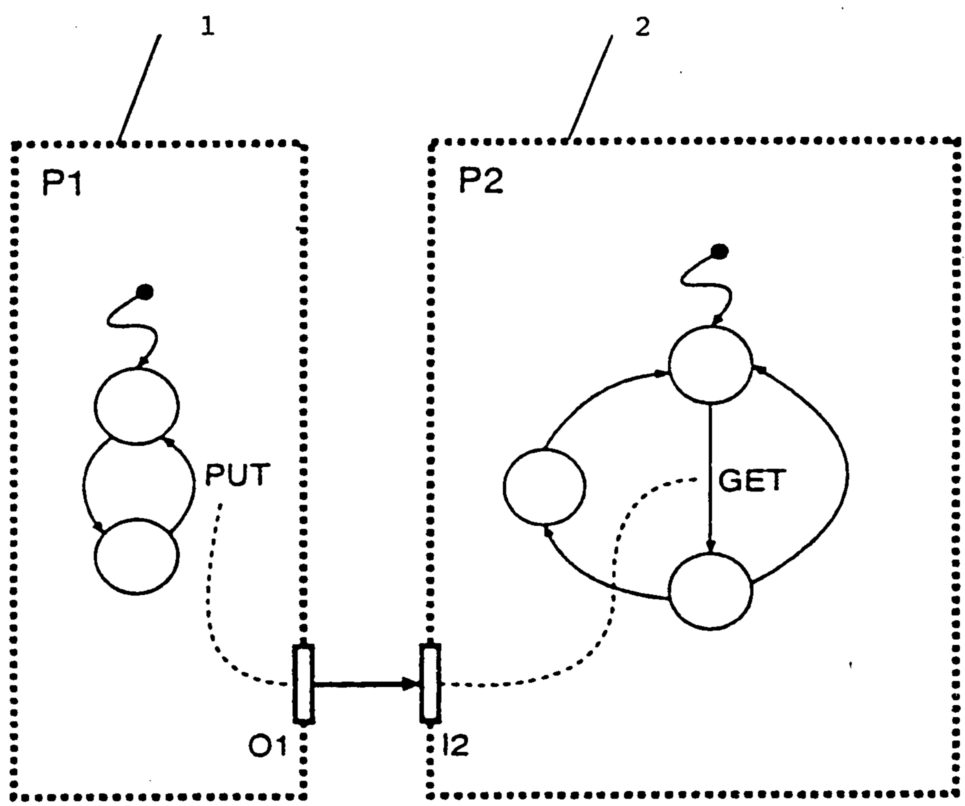


Figure 1

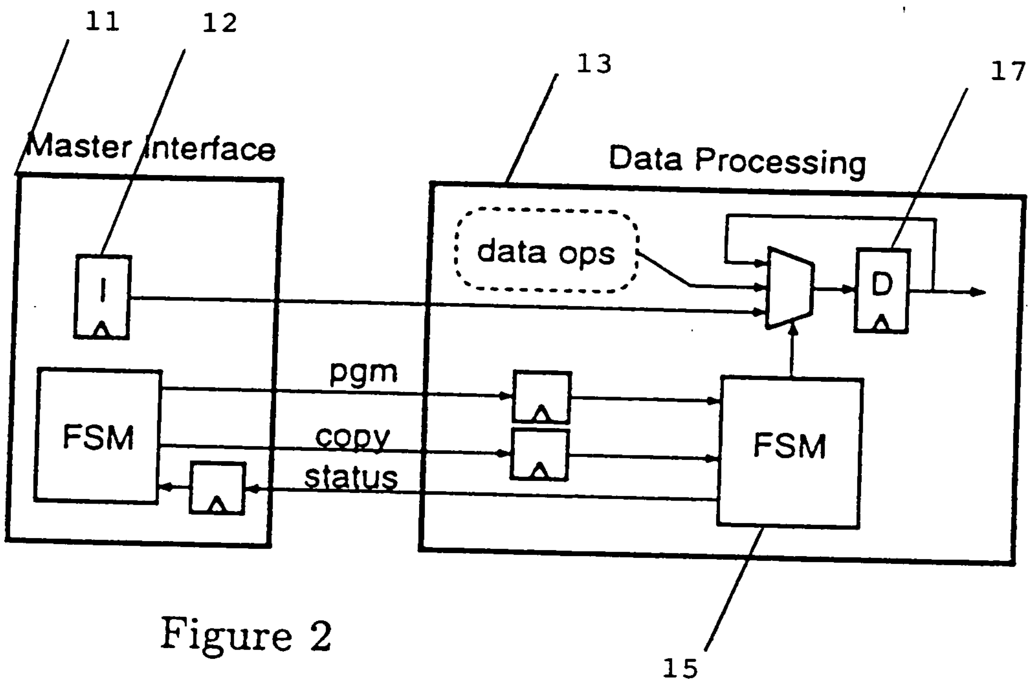

11
12
13
17
Master Interface
Data Processing
data ops
I
D
pgm
copy
status
FSM
FSM
15

Figure 2

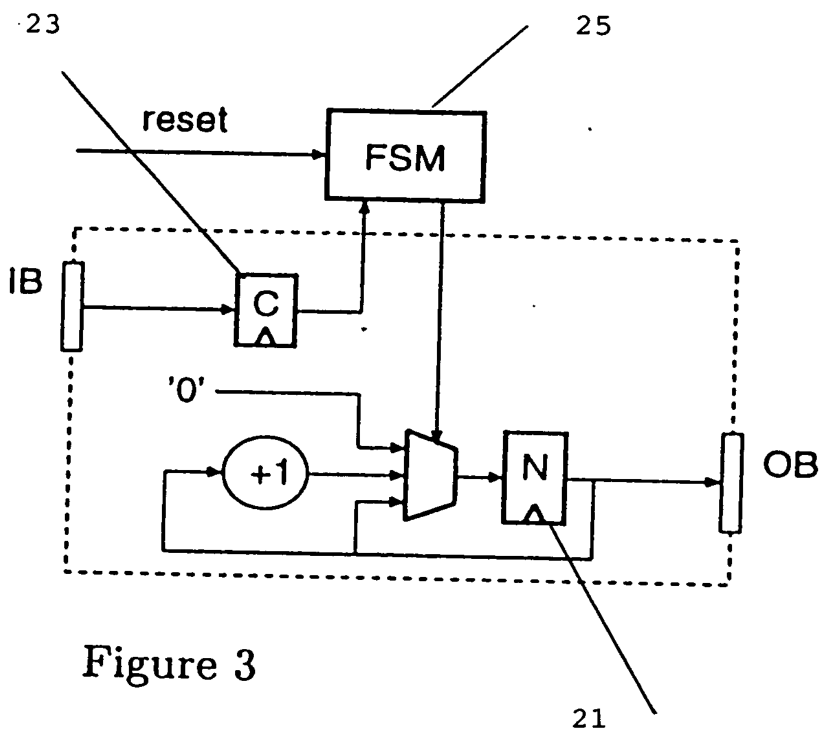
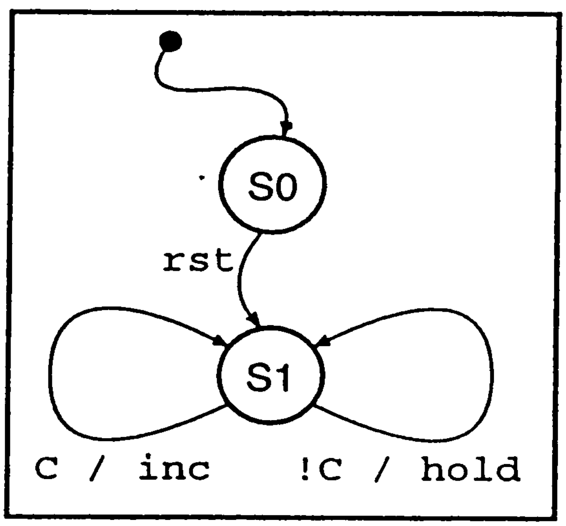


Figure 3

FSM



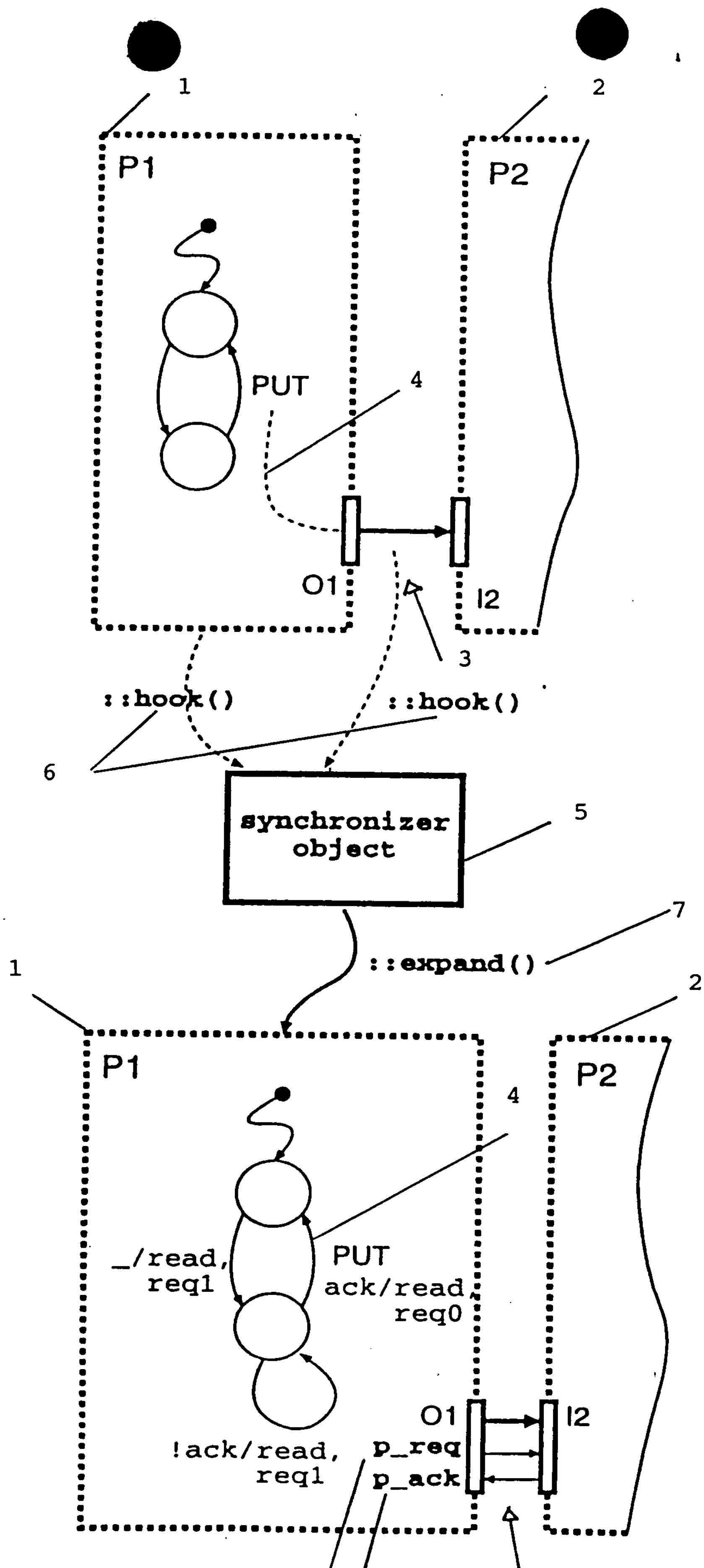

```
register C;
register N;
input    IB;
output   OB;

rst:     N = 0;
         C = IB;

inc:     N = N + 1;
         C = IB;
         OB = N;

hold:    C = IB;
         OB = N;
```

Figure 4



1
2
P1
P2
PUT
4
O1
I2
3
::hook()
::hook()
6
synchronizer object
5
::expand()
7
1
2
P1
P2
4
_/read, req1
PUT ack/read, req0
!ack/read, req1
O1
I2
p_req
p_ack
3
8
9

Figure 5

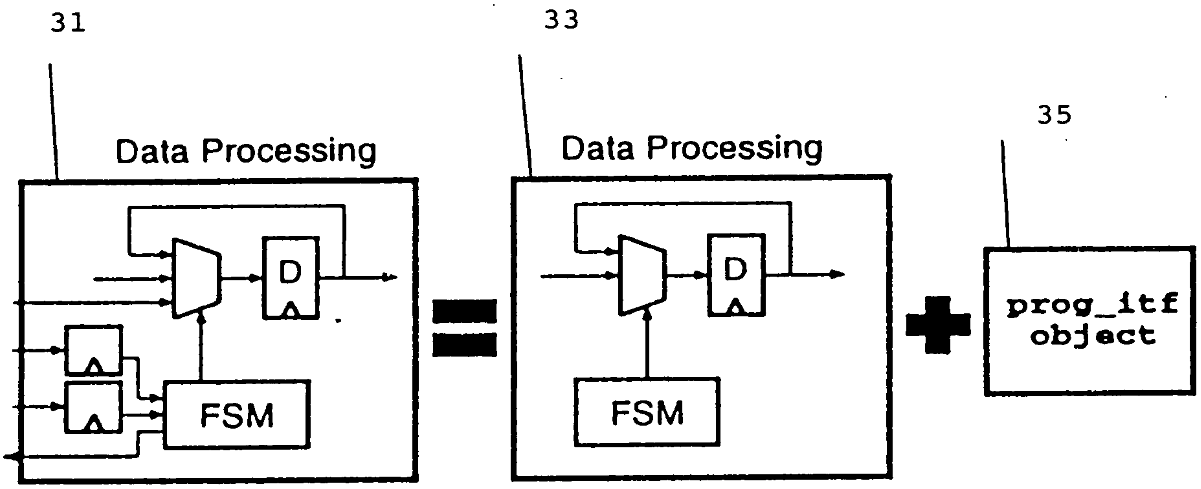

31
33
35
Data Processing
Data Processing
D
D
FSM
FSM
prog_itf
object

Figure 6

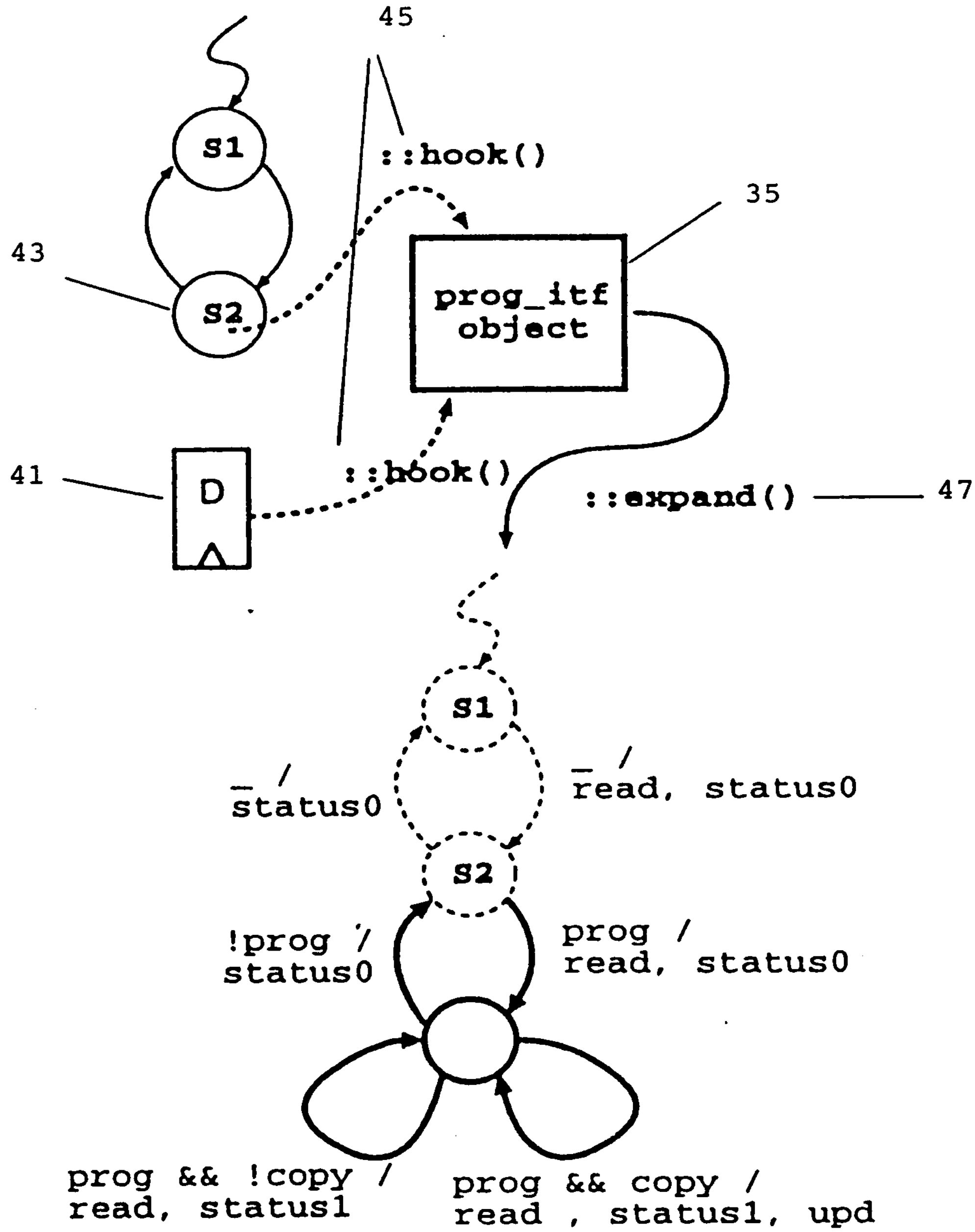

45
S1
::hook()
35
43
S2
prog_itf
object
41
D
::hook()
::expand()
47
S1
_ /
status0
_ /
read, status0
S2
!prog /
status0
prog /
read, status0
prog && !copy /
read, status1
prog && copy /
read , status1, upd

Figure 7

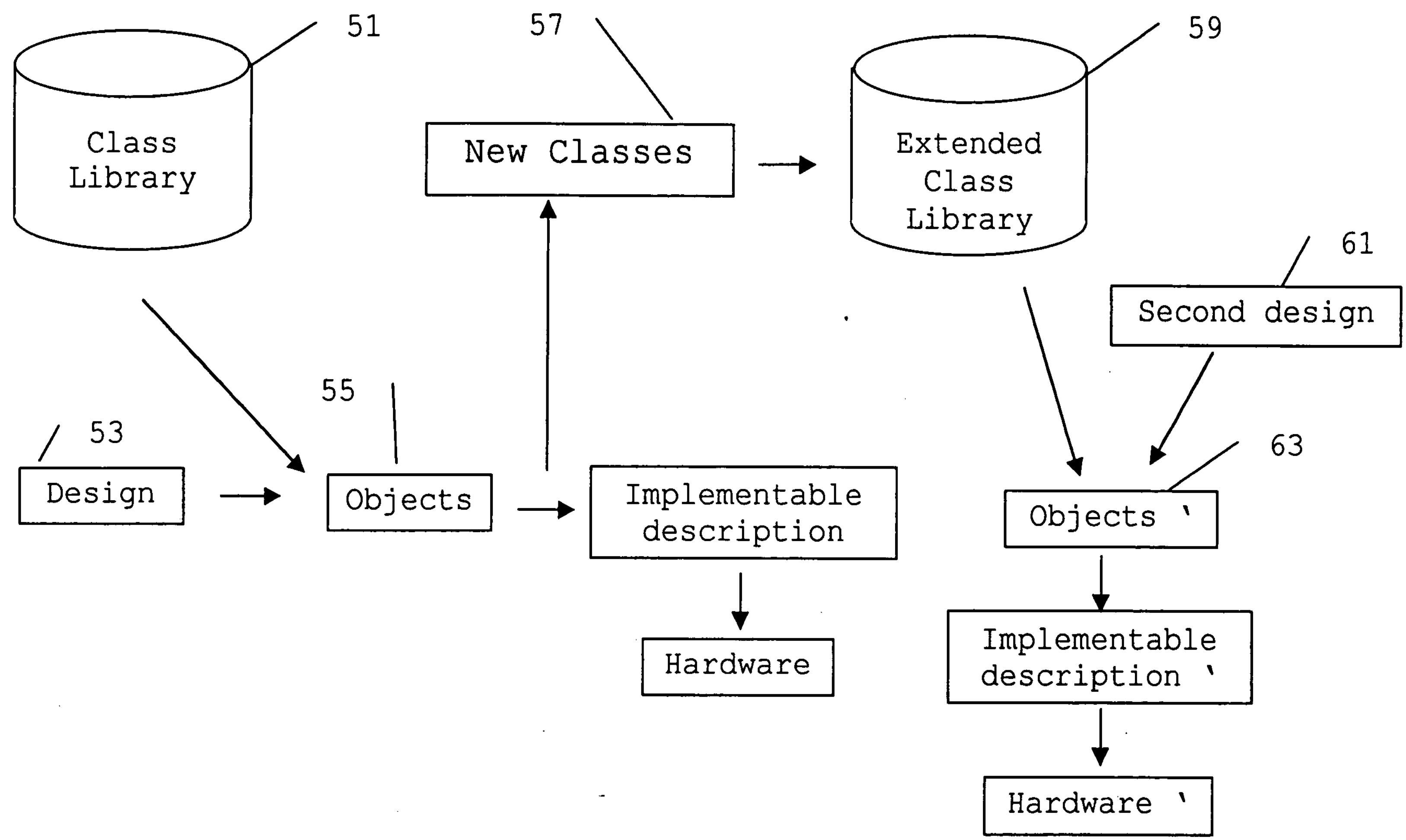

51
57
59
Class Library
New Classes
Extended Class Library
61
Second design
53
55
63
Design
Objects
Implementable description
Objects '
Hardware
Implementable description '
Hardware '

Figure 8